REMARKS

Applicants acknowledge receipt of the Examiner's Office Action dated September 19, 2007.

Rejection of Claims under 35 U.S.C. §102

Claims 1, 4-6, 8-10, 12, 15-17 and 19-21 stand rejected under 35 U.S.C. §102(e) as being anticipated by Shackelford *et al.*, U.S. Publication No. 2005/0154845 (*Shackelford*). *Shackelford* has an effective date of January 9, 2004, which is the purported date of filing of the application that led to that reference. Under 35 U.S.C. §102(e), in order to preclude patentability, a published application such as *Shackelford* must have an effective date prior to the invention by the Applicants for patent. Applicants can establish invention prior to a reference by providing evidence of reduction to practice prior to the effective date of the reference. *See* 37 C.F.R. §1.131(b). Applicants respectfully submit that the invention disclosed and claimed in the present Application was reduced to practice prior to the effective date of *Shackelford* and is therefore allowable over *Shackelford*.

Attached hereto as Exhibit A is a redacted copy of an Invention Disclosure Form that led to the filing of the present Application. As indicated on the face of Exhibit A and affirmed on the declarations of Ronald S. Karr and Anand A. Kekre submitted herewith as Exhibits B and C, respectively, the Invention Disclosure Form was submitted by inventors Ronald S. Karr and Anand A. Kekre on or before October 16, 2003 and was directed toward "**Coherently sharing any form of "instant" snapshot separately from base volumes**". *See* Exhibit A, p.1 (compare with title of present Application: "COHERENTLY SHARING ANY FORM OF INSTANT SNAPSHOTS SEPARATELY FROM BASE VOLUMES"). The Invention Disclosure Form further describes the invention as follows:

Distributed block virtualization basically distributes a description of how a virtual block device (for example, a logical volume or a virtual LUN) relates to underlying storage, as well as how distributed block virtualization components might relate in order to accomplish certain functions.

Block virtualization components can live in various places. They can be software layers on servers that run file systems or databases. They can be in the disk arrays. They can be in intermediate devices residing in storage networks between hosts and disk arrays. They can be in the host bus adapters that connect a host to a network. Any of these components could participate in some form of distributed virtualization. It is also possible that any of these components could implement a non-distributed virtualization even though other components do implement distributed virtualization. Components can cooperate to provide distributed virtualization either across a layer or between layers, or layered components can implement isolated virtualization layers where a higher layer uses a lower layer in the same way the upper layer would simply use a disk drive.

A typical implementation of distributed block virtualization will have components running on some collection of nodes to identify and organize some collection of underlying logical disks (*device management*). Additionally, they will have *configuration management* components that store and retrieve information that describes the logical structure associated with the virtualization (how logical and physical disks are divided, aggregated, mirrored, and so on). Various *distributed virtualization coordination* components analyze the logical structure and communicate information about that structure to the various *I/O engines* that receive and transform read and write requests (the incoming I/O stream) to virtual block devices and convert them into network messages and reads and writes to underlying logical and physical disk drives.

The configuration associated with a virtual block device may change over time, such as to add or remove mirrors; migrate data to new storage; increase or decrease the size of the device; create, manipulate, or remove snapshots; add structure for a new capability; etc. Changes that affect several I/O engines must often result in coherent updates that are effectively *atomic* relative to some aspect of the I/O stream. This is generally done through some kind of distributed transactional update, coupled with some form of *I/O quiesce*, which temporarily blocks activity while distributed structures are updated.

Figure 1 graphically depicts a typical logical volume-based block virtualizer, such as that implemented by SANVM. A coordinator, in SANVM called a *volume server*, serves the configuration management and distributed virtualization coordination functions. The SANVM volume server reads and updates configuration information from VxVM configuration databases (which are actually stored on the disks that are being virtualized) in persist and retrieve the logical configuration. The logical configuration for a single volume is depicted in figure 1 as a tree of virtual objects. This tree is coherently communicated and updated to all the nodes that share access to

> **that volume. Each node can then independently use that tree of virtual objects to transform the I/O stream coming into the I/O engine on that node into I/O requests directly to the physical disks that underlie the volume. SANVM also has another software component (called a *log coordinator*) that is an example of a special component that coordinates some virtualization function. The log coordinator in SANVM coordinates update activity associated with recoverable mirroring and snapshots. Such *function coordinator* components are likely to be prevalent within many distributed virtualization implementations.**

Exhibit A, p.2. Thus, Exhibit A provides disclosure of the current claims.

Exhibit A further indicates that the claimed invention was reduced to practice at least prior to the writing of Exhibit A. Question 12 asks "Has the invention been built or implemented?" In response, Exhibit A provides "**It is in CVM 4.0 and will be in SANVM.**" Exhibit A thus establishes that the disclosed problem solution that led to the present Application was reduced to practice prior to the October 16, 2003 date of Exhibit A.

In light of the above-described evidence provided by Exhibit A that the claimed invention was reduced to practice prior to the effective date of *Shackelford*, Applicants submit that Claims 1, 4-6, 8-10, 12, and 15-17 and 19-21 are allowable over *Shackelford*. Applicants therefore respectfully request the Examiner's reconsideration and withdrawal of the rejections as to these claims and an indication of the allowability of same.

Rejection of Claims under 35 U.S.C. § 103

Claims 2, 3, 11, 13, 14, and 22 are rejected under 35 U.S.C. §103(a) as being obvious over *Shackelford* in view of U.S. Patent No. 6,222,558 to Berg (*Berg*). Claims 2, 3, 11, 13, 14, and 22 depend from and further patentably distinguish independent claims 1, 6, 12 and 17, respectively, and, for at least that reason, Applicants respectfully submit that they are allowable. Applicants therefore respectfully request the Examiner's reconsideration and withdrawal of the final rejections as to these claims and an indication of the allowability of same.

CONCLUSION

Applicants submit that all claims are now in condition for allowance, and an early notice to that effect is earnestly solicited. Nonetheless, should any issues remain that might be subject to resolution through a telephonic interview, the Examiner is requested to telephone the undersigned.

If any extensions of time under 37 C.F.R. §1.136(a) are required in order for this submission to be considered timely, Applicant hereby petitions for such extensions. Applicant also hereby authorizes that any fees due for such extensions or any other fee associated with this submission, as specified in 37 C.F.R. §1.16 or §1.17, be charged to Deposit Account 502306.

Respectfully submitted,

Eric A. Stephenson
Attorney for Applicants
Reg. No. 38,321
Telephone: (512) 439-5093
Facsimile: (512) 439-5099